UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte HANDONG WU,
JEROME FREEDMAN,
and CHRISTOPHER J. IVORY

Appeal 2007-0836
Application 10/091,645[1]
Technology Center 2100

Decided: 18 March 2008

Before KENNETH W. HAIRSTON, LEE E. BARRETT, and
JAY P. LUCAS, *Administrative Patent Judges.*

BARRETT, *Administrative Patent Judge.*

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134(a) from the final rejection of claims 1-8, 10, 11, 14-16, and 18-22. We have jurisdiction pursuant to 35 U.S.C. § 6(b).

We affirm-in-part.

---

[1] Application for patent filed March 5, 2002, entitled "Network Intrusion Detection and Analysis System and Method."

BACKGROUND

The claims are directed to an intrusion detection and analysis system and method having a data monitoring device separate from an intrusion detection device. The intrusion detection device accesses applications of the data monitoring device using application program interfaces (APIs).

Claim 1 is illustrative:

1. An intrusion detection and analysis system comprising:

a data monitoring device comprising a capture engine operable to capture data passing through the network in response to a trigger and configured to monitor network traffic, decode protocols for grouping packets into different protocol presentations and assembling the packets into high level protocol groups, and analyze received data for managing the network by collecting statistics, and detecting broken lines, traffic loads, and network errors;

an intrusion detection device separate from the data monitoring device, the intrusion detection device comprising a detection engine operable to perform intrusion detection on data provided by the data monitoring device;

application program interfaces configured to allow the intrusion detection device access to applications of the data monitoring device to perform intrusion detection; and

memory for storing reference network information used by the intrusion detection device to determine if an intrusion has occurred;

wherein the application program interfaces allow the intrusion detection device to leverage the separate data monitoring device, by allowing the intrusion detection device to call an application program

interface configured to open a protocol decoding application associated with the separate data monitoring device, and by allowing the intrusion detection device to call an application program interface configured to open an alarm generation application associated with the separate data monitoring device.

THE REFERENCES

Vaidya	US 6,279,113 B1	Aug. 21, 2001
Porras	US 2003/0101358 A1	May 29, 2003 (filed Nov. 28, 2001)

THE REJECTIONS

Claims 21 and 22 stand rejected under 35 U.S.C. § 112, first paragraph, for lack of enablement of the claimed APIs.

Claims 21 and 22 stand rejected under 35 U.S.C. § 112, second paragraph, as failing to particularly point out and distinctly claim the invention.

Claims 1-8, 10, 11, 14-16, 18, and 19 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Vaidya or, in the alternative, under § 103(a) as being unpatentable over Vaidya and Porras.[2]

[2] The statement of the rejection in the Examiner's Answer refers to claims 1-19, but claims 9, 12, 13, and 17 have been canceled.

DISCUSSION

35 U.S.C. § 112, first paragraph

The Examiner concludes that the specification does not enable the particular APIs in the claims, e.g., "frame_context_pointer_position" in claim 21. Appellants argue page 12 of the specification defines the forms that the APIs may take (Br. 10) and indicates that the APIs are used to parse, generate, and load signatures, and other functions (Reply Br. 2).

"The test of enablement is whether one reasonably skilled in the art could make or use the invention from the disclosures in the patent coupled with information known in the art without undue experimentation." *United States v. Telectronics, Inc.*, 857 F.2d 778, 785 (Fed. Cir. 1988). The factors to be considered in determining whether a disclosure would require "undue experimentation" are summarized in *In re Wands*, 858 F.2d 731, 737 (Fed. Cir. 1988): (1) the quantity of experimentation necessary; (2) the amount of direction or guidance presented; (3) the presence or absence of working examples; (4) the nature of the invention; (5) the state of the prior art; (6) the relative skill of those in the art; (7) the predictability or unpredictability of the art; and (8) the breadth of the claim. The *Wands* factors "are illustrative, not mandatory. What is relevant depends on the facts." *Amgen, Inc. v. Chugai Pharm. Co., Ltd.*, 927 F.2d 1200, 1213 (Fed. Cir. 1991).

In this case we find there is no guidance as to the function or scope of the claimed APIs that would allow one skilled in the art to make them. Appellants fail to show that the claimed APIs are conventional items in the

art or that they have ordinary and customary meanings in the art so that one of ordinary skill in the art would know how to make them. The rejection of claims 21 and 22 under 35 U.S.C. § 112, first paragraph, is affirmed.

35 U.S.C. § 112, second paragraph

The Examiner concludes that the particular named APIs fail to define the invention. Appellants argue that this is just a blanket assertion without any specifics (Br. 10) and rely on the arguments made for enablement (Reply Br. 3)

"The test for definiteness is whether one skilled in the art would understand the bounds of the claim when read in light of the specification. If the claims read in light of the specification reasonably apprise those skilled in the art of the scope of the invention, § 112 demands no more. The degree of precision necessary for adequate claims is a function of the nature of the subject matter." *Miles Laboratories, Inc. v. Shandon Inc.*, 997 F.2d 870, 875 (Fed. Cir. 1993) (citations omitted).

The names for the APIs do not describe their function or operation and, thus, it is unknown what is meant by these terms. Appellants fail to show that the API names have well-known meanings in the computer art such that the meanings are definited. Accordingly, the rejection of claims 21 and 22 under 35 U.S.C. § 112, second paragraph, is affirmed.

35 U.S.C. § 102

The Examiner finds that "Vaidya does not specifically refer to an application program interface to access the data and functionalities of the data-monitoring device, but Porras clearly suggests the use of APIs . . . in development of Intrusion Detection Systems" (Ans. 7). The Examiner does not contend that an API is necessarily inherent in Vaidya. Since an "application program interface" is recited in each of independent claims 1, 11, and 19, and since an API is not asserted to be expressly disclosed or inherent, the anticipation rejection claims 1-8, 10, 11, 14-16, 18, and 19 over Vaidya is reversed.

35 U.S.C. § 103(a)

The Examiner holds that "[i]t would have been obvious to a person skilled in the art to use APIs, as clearly disclosed by Porras, as a means to transfer information between objects or elements of a distributed system in order to build the intrusion detection system invented by Vaidya" (Ans. 7).

An "application program interface (API)" is defined as "A functional interface supplied by the operating system or by a separately orderable licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or the licensed program," George McDaniel, *IBM Dictionary of Computing* 28 (McGraw-Hill, Inc. 1994). An API is a set of standardized requests that have been defined for the program being called upon to perform some function. "In essence, a program's API defines the proper way for a

developer to request services from that program." *QuickStudy: Application Programming Interface (API)* (http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=43487&pageNumber=1) Computerworld (Jan. 10, 2000). We agree with the Examiner that the use of APIs to communicate between programs was notoriously well known in the computer software art at the time of the invention and that it therefore would have been obvious to use APIs to interface between programs for the known advantages of APIs. Appellants do not contest that the use of APIs to communicate between programs would have been obvious, but argue that specific claim limitations are not suggested; e.g., "appellant does not merely claim using APIs, but instead specifically claims 'allowing the intrusion detection device to call an application program interface configured to open a protocol decoding application associated with the separate data monitoring device, and . . . allowing the intrusion detection device to call an application program interface configured to open an alarm generation application associated with the separate data monitoring device" (Reply Br. 8-9). Thus, the question is whether the references disclose the communication functions between a separate "data monitoring device" and an "intrusion detection device," which interfaces can be implemented using APIs.

Appellants argue that Vaidya does not teach the claimed "intrusion detection device separate from the data monitoring device." Although not clearly stated, we understand Appellants' argument to be that separate "devices" means separate pieces of hardware, such as separate processors.

The Examiner states that: (1) separate devices only requires separate functionalities (Final Rejection 3); (2) Appellants' Figure 2 shows a single element, the network intrusion detection and analysis system (NIDAS) 18, which contains both the intrusion detection device and the data monitoring device, so separation makes no difference in functionality (Final Rejection 3); (3) Vaidya does not limit his invention to one processor only because Figure 4 shows separate modules 34, 36, and 38 for performing separate functionalities (Final Rejection 4); (4) the register cache 40 is shown as a separate device within the virtual processor 36 in Vaidya (Final Rejection 4); (5) Vaidya's claim 1 recites separate steps of monitoring and determining network intrusion (Final Rejection 4-5); (6) Appellants' Figure 1 is a functional block diagram and hence the separate devices imply only functional separation; (7) the NIDAS 18 in Appellants' Figure 2 is a single element which implies that only functional separation is required and, moreover, functional separation is not excluded by the claim language (Ans. 22); (8) a processor is an element capable of processing data and performing operations (Ans. 22); and (9) elements 34, 36, and 38 in Vaidya are different processors performing different operations (Ans. 22).

Appellants argue: (1) Figure 1 and the specification refer to separate *devices*, not just separate functionalities (Br. 11; Reply Br. 3); (2) this argument is not addressed in the Brief; (3) Figure 4 of Vaidya shows separate modules, not separate processors, and the only processing device in Vaidya is the virtual processor 36 (Br. 11; Reply Br. 4-5); (4) the register

cache only temporarily stores data extracted from a packet and is not a "data monitoring device," as claimed, so the functionality of items 36 and 40 does not meet the claim language (Br. 11); (5) merely claiming separate steps does not meet the limitation of separate "devices" (Br. 12; Reply Br. 7); (6) Figure 1 shows two separate devices (Reply Br. 3); (7) the fact that the NIDAS includes a separate intrusion detection and a separate network analysis and data monitoring device suggests that the separation is not just functional (Reply Br. 4); (8) Vaidya discloses a processor as a software based virtual processor or a hardware based processor (Reply Br. 5); and (9) the mere disclosure of separate modules does not suggest separate devices (Reply Br. 5). Appellants make numerous other arguments about separate devices in the Reply Brief, which we interpret to mean that the devices are separate and distinct pieces of hardware.

The term "device" is a generic term that does not connote structure and is treated the same as "means." *See Mass. Inst. of Tech v. Abacus Software*, 462 F.3d 1344, 1354 (Fed. Cir. 2006). The specification does not define what is meant by a "device." The structure disclosed in the specification for performing the functions of the "intrusion detection device" and the "data monitoring device" is a computer programmed with software. As noted by the Examiner, both functions can be performed by the same computer as shown by the NIDAS 18 in Figure 2. The structure of each "device" is the computer plus the software, so one computer can include several "devices." Appellants argue that there are separate devices without

ever addressing the relevant question of what separate structures correspond to the "devices." The functional blocks in Figure 1 do not imply separate physical processors. Appellants argue that Vaidya only discloses one processor and so does not have separate "devices," implying that a device is a processor, without saying where the specification discloses a "separate device" to be a separate processor. Thus, we agree with the Examiner that separate software modules executing on the same computer are separate "devices." However, it is required that the functionality be "separate" so that one can identify a "data monitoring device" separate from an "intrusion detection device" and that the intrusion detection device can communicate to open applications in the data monitoring device as discussed *infra*.

Appellants argue that Vaidya does not perform the function of the "data monitoring device." The Examiner refers to the register cache element 40 in Figure 4 of Vaidya and the disclosure of the data collector 10 which functions to monitor network data (referring to col. 5, ll. 13-15; col. 6, l. 57 to col. 7, l. 24; col. 8, ll. 15-40; col. 9, ll. 4-21) (Ans. 5-6). Appellants argue that "such register cache that only stores information from data packets does not meet appellant's claimed 'data monitoring device,' which specifically 'capture[s] data passing through the network,' 'monitor[s] network traffic,' 'decode[s] protocols for grouping packets into different protocol presentation and assembling the packets into high level protocol groups,' and 'analyze[s] received data,' in the manner claimed by appellant" (Reply Br. 5-6).

The data collector 10 monitors network data to detect packets addressed to network objects on a network segment (col. 6, ll. 57-59) and therefore is a "data monitoring device." (As will be discussed, the data collector is also an "intrusion detection device.") The portions of Vaidya noted by the Examiner reasonably disclose that the data collector 10 is "operable to capture data passing through the network in response to a trigger and configured to monitor network traffic, decode protocols for grouping packets into different protocol presentations and assembling the packets into high level protocol groups, and analyze received data," as recited in claim 1, and Appellants provide no *reasons* why it does not. Although not expressly discussed, it appears that the Examiner is treating the limitation "for managing the network by collecting statistics, and detecting broken lines, traffic loads, and network errors" as a statement of intended use as indicated by the word "for" because the Examiner states that "Vaidya's system is ***capable of*** collecting statistics and detecting broken lines, traffic loads and network errors" (emphasis added) (Ans. 6). The "intended use" of a machine is not germane to the issue of patentability of the machine itself. *In re Casey*, 370 F.2d 576, 580, (CCPA 1967). There is an extensive body of precedent on the question of whether a statement in a claim of purpose or intended use constitutes a limitation for purposes of patentability. *See generally Kropa v. Robie*, 187 F.2d 150, 155-59 (CCPA 1951) and the authority cited therein, and cases compiled in 2 Chisum, *Patents* § 8.06[1][d] (2006). Such statements often, although not

necessarily, appear in the claims preamble. *In re Stencel*, 828 F.2d 751, 754 (Fed. Cir. 1987). However, the structure must be capable of performing the intended use. It is not apparent that the data collector 10 in Vaidya is capable of performing the functions of "collecting statistics, and detecting broken lines, traffic loads, and network errors" without being specially programmed. Nevertheless, while it would have been more prudent to show a data monitoring system that provides these functions, Appellants do not seem to deny that "collecting statistics, and detecting broken lines, traffic loads, and network errors" were well-known data monitoring functions. Accordingly, we conclude that these functions of a data monitoring device are taught by Vaidya or at least would have been obvious to one skilled in the art.

The main issue appears to be the limitations of a "data monitoring device" separate from an "intrusion detection device," "application program interfaces configured to allow the intrusion detection device access to applications of the data monitoring device to perform intrusion detection" and "wherein the application program interfaces allow the intrusion detection device to leverage the separate data monitoring device, by allowing the intrusion detection device to call an application program interface configured to open a protocol decoding application associated with the separate data monitoring device, and by allowing the intrusion detection device to call an application program interface configured to open an alarm generation application associated with the separate data monitoring device."

Appellants argue that the intrusion detection device "is allowed to leverage the separate functionality such that the intrusion detection device is allowed to leverage the separate data monitoring device" (Br. 13; Reply Br. 9).

Although we have concluded that "separate devices" does not require separate and distinct pieces of hardware, and could be modules with separate functionalities on the same computer, and that it would have been obvious to use APIs to interface between programs, it is still necessary to identify separate functional units that correspond to the claimed "data monitoring device" and "intrusion detection device" and identify interfaces where the intrusion detection device calls a "protocol decoding application" and an "alarm generation application" associated with the separate data monitoring device. The term "leverage" is interpreted to mean only the natural result of the subsequently recited limitations.

Vaidya has functions corresponding to the "data monitoring device" and the "intrusion detection device," but these functions are part of one overall system and there is no clear line of demarcation between separate functionalities whereby the interface between the devices can be determined. That is, we are not able to clearly partition the elements and descriptions in Vaidya into a separate "data monitoring device" and "intrusion detection device," nor identify interfaces where the intrusion detection device calls a "protocol decoding application" and an "alarm generation application" associated with the separate data monitoring device. Figures 2 and 4 of Vaidya come the closest to showing the claimed elements and interfaces.

The virtual processor 36 performs the function of both a "data monitoring device" (col. 7, ll. 18-24) and "intrusion detection device" (col. 7, ll. 34-36). The register cache 40 only temporarily stores information for use by the virtual processor (col. 7, ll. 15-24) and is not one of the two devices. There is no indication of an interface from the "intrusion detection device" part to a "protocol decoding application" and an "alarm generation application" associated with a separate "data monitoring device" part. While there is a connection/interface between the virtual processor 36 and a reaction module 38 in Figures 2 and 4, and while the reaction module can be considered an "alarm generation application," there is no disclosure of the reaction module being associated specifically with the "data monitoring device." Vaidya does not describe the claimed organization of functions, i.e., a separate "intrusion detection device" and a separate "data monitoring device" which communicate through an interface of some sort, so, although we agree that it would have been obvious to use APIs for the interface, the claimed subject matter is not met. Independent claims 1, 11, and 19 all require the discussed limitations. Accordingly, we reverse the obviousness rejection of claims 1-8, 10, 11, 14-16, 18, and 19.

CONCLUSION

The rejection of claims 21 and 22 under 35 U.S.C. § 112, first paragraph, is affirmed.

The rejection of claims 21 and 22 under § 112, second paragraph, is affirmed.

The rejection of claims 1-8, 10, 11, 14-16, 18, and 19 under § 102(e) is reversed.

The rejection of claims 1-8, 10, 11, 14-16, 18, and 19 under § 103(a) is reversed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv) (2006).

AFFIRMED-IN-PART

Zilka-Kotab, PC
P.O. BOX 721120
SAN JOSE, CA 95172-1120